When Machines Evaluate Machines: A Critical Assessment of AI Evaluators in the Context of Optimizing LLaMa-Based Retrieval-Augmented-Generation Pipelines

Team: A-20, Darpan Beri

28th March 2025

Contents

Artificial Intelligence Usage Disclaimer

Portions of this report benefited from the assistance of Large Language Models (LLMs). Specifically, AI tools were utilized for tasks including grammar checking, debugging code associated with the experimental setup, and generating code snippets for data visualizations presented herein. The core research, analysis, interpretation of results, and the substantive writing remain the work of the author. For any concerns and inquiry, feel free to reach out to the author at darpanberi.99@gmail.com.

1. Introduction

The rapid advancement of Large Language Models (LLMs) marks a significant step towards the long-held goal of human-like machine communication [1], [2]. These cutting-edge creations, also termed as artificial intelligences (AI) are capable of understanding and generating human-like text. They are being increasingly integrated into organizations with gusto, with sizeable financial investment being made into infrastructure around these LLMs [3]. However, problems with LLMs remain. LLMs are limited by static knowledge cutoffs inherent in their training data and can find it difficult to recall information that was less frequent (long-tail) in their training data [4], [5]. They often struggle with factual accuracy, generating "hallucinations", outputs which are inconsistent with known reality [6]. Furthermore, retraining LLMs with updated information remains computationally expensive and resource-intensive [7].

Retrieval-Augmented Generation (RAG) offers a compelling architectural solution to mitigate these issues [8]. By design, RAG pipelines enable LLMs to access and incorporate relevant information from external, up-to-date knowledge sources in real-time before generating a response. This process not only enhances the accuracy and relevance of the output but also allows for source attribution, increasing transparency and user trust [9]. RAG frameworks are particularly valuable for organizations seeking to leverage LLMs with proprietary data without costly retraining cycles [8], [9].

Research Problem and Objectives

Despite the promise of RAG, its performance is sensitive to configuration choices, particularly how external knowledge is processed and presented to the LLM. The internal configuration settings of models like RAG and LLMs are known as hyperparameters. Key

hyperparameters include chunk size, which dictates the length of text segments retrieved from the knowledge base, and top_k, which determines the number of most relevant segments passed to the LLM. Table 1 provides an example of how the chunk_size and top_k hyperparameters influence the data that is provided to the LLM within the RAG pipeline. Optimizing these parameters is crucial, yet their combined impact, especially within specific model families like LLaMa, requires further investigation.

Chunk_size (number of words in a chunk)	**Top_k (number of chunks)**	**Retrieved Chunks (Visible to LLM within the RAG model)**
50	2	Chunk 1 (50 words): "The mitochondrion is the powerhouse of the cell. It generates ATP through cellular respiration..." Chunk 2 (50 words): "ATP is the primary energy currency of the cell, fueling numerous biological processes including muscle contraction..."
100	1	Chunk 1 (100 words): "The mitochondrion is the powerhouse of the cell. It generates ATP through cellular respiration. ATP is the primary energy currency of the cell, fueling numerous biological processes including muscle contraction..."

Table 1: Impact of chunk_size and top_k on Retrieved Context in a RAG Pipeline

Furthermore, evaluating the quality of RAG outputs presents its own challenges. While using another LLM as an automated evaluator seems scalable [10], [11], its reliability compared

to human judgment is a critical concern, particularly for deployment in real-world applications where accuracy is paramount.

This research aims to answer the following questions:

- How do chunk size and top_k interact to influence the semantic accuracy of RAG-generated answers?
- How reliable is the automated LLM evaluator compared to human assessment under different RAG configurations?

Methodological Context and Evolution

To answer these questions, this study utilized a RAG pipeline built with a LLaMa 3.1 8B model [12]. This specific Llama model instance was run in a memory-saving mode ("8-bit quantization") due to hardware limitations. The system used information from Wikipedia as its knowledge source and was tested using a standard question-answering dataset [12], [13]. Initially, a smaller LLaMa 3.2 1B model was intended for automated evaluation (judging semantic equivalence between generated and ground-truth answers). However, this model proved unreliable, often failing to adhere to the "Yes/No" output format. Consequently, the evaluator was upgraded to another LLaMa 3.1 8B instance. While the instructions following improved, preliminary analysis of the 40 experimental runs (covering all chunk_size/top_k combinations across 918 questions each) revealed persistent inaccuracies and inconsistencies in the automated evaluations.

This critical finding necessitated a shift in methodology. To establish a reliable baseline for RAG performance and rigorously assess the evaluator's fidelity, a subset of 1200 outputs (30 fixed random samples from each of the 40 experiments) was manually evaluated by a human

researcher using the same semantic equivalence criterion. This dual approach allows not only for the analysis of hyperparameter effects on RAG quality (based on human evaluation) but also for a quantitative comparison between automated and human judgments, including a logistic regression analysis to explore factors influencing evaluator accuracy.

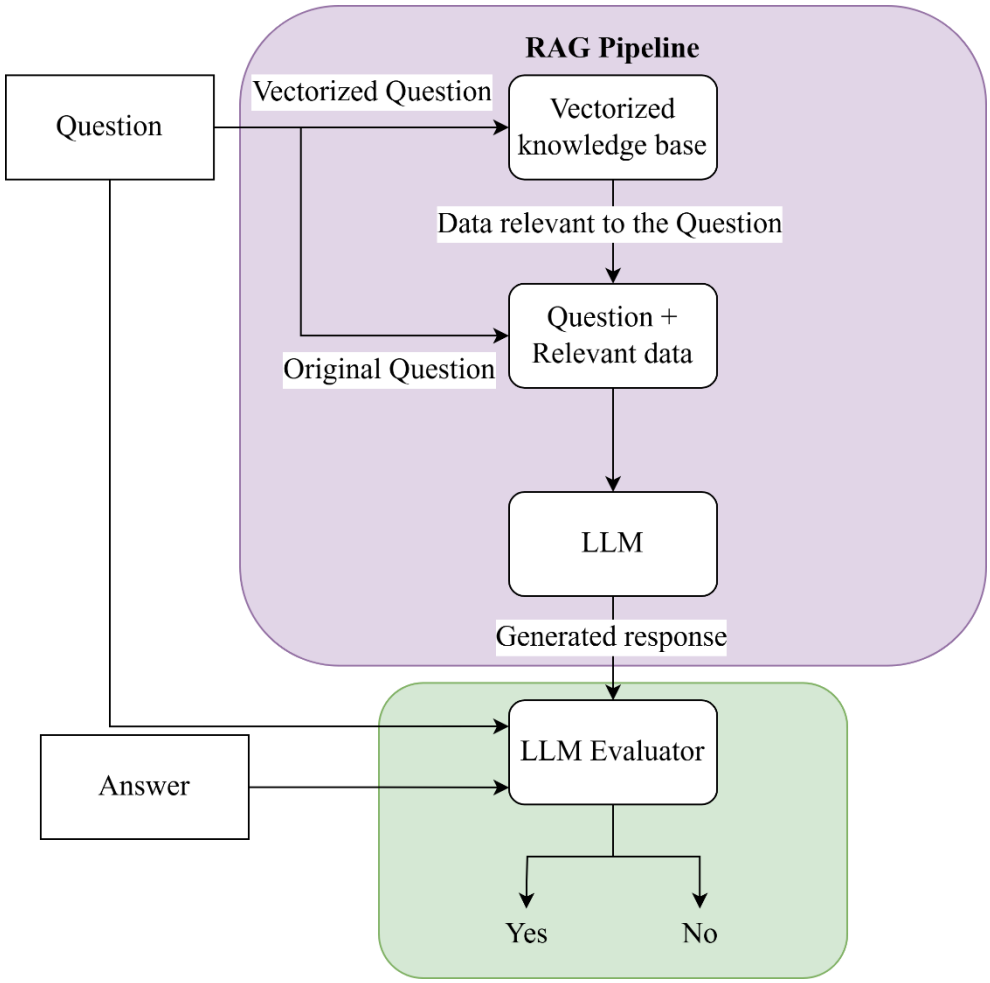

Figure 1: Overview of the RAG and LLM Evaluator for this study. The system gathers and organizes relevant background information from a pre-processed, vectorized knowledge base. It then retrieves the most pertinent pieces of this information and combines them with the question to create a context-rich prompt. The large language model (LLM) generates an answer based on

this prompt, after which an automated evaluator checks whether the response is factually and semantically correct.

Motivation and Rationale

The primary motivation for this project arises from the practical challenges and opportunities presented by deploying Large Language Models (LLMs) in real-world commercial and research contexts. While LLMs offer unprecedented capabilities, ensuring their reliability is paramount. Issues like hallucination and knowledge limitations hinder user trust and adoption. Retrieval-Augmented Generation (RAG) pipelines offer a promising solution by grounding LLM responses in verifiable external information, thereby enhancing accuracy and user confidence. Optimizing these RAG pipelines, specifically through careful tuning of parameters like chunk size and top_k, is crucial for maximizing their effectiveness. Furthermore, robust RAG systems can be seen as foundational components for more complex future applications, potentially including the development of "agentic AI" systems designed for end-to-end task automation, where reliability will be even more critical [14].

The expanded focus addresses a key bottleneck in scaling LLM deployment: the automation of evaluation. While using LLMs to evaluate other AI outputs promises significant resource efficiencies compared to labor-intensive human evaluation, it raises fundamental questions about oversight and accuracy. As aptly questioned by IBM in their article on the subject, "Who watches the AI watchers?" [15]. The potential for bias, error, or misalignment in automated evaluations necessitates scrutiny. This study directly confronts this challenge by empirically comparing the LLM evaluator's judgments with human evaluations, shedding light on the practical implications of relying on AI for assessing AI quality. Simultaneously, the

project retains its original goal of investigating the combined impact of chunk_size and top_k, addressing a gap where much existing literature, like Wang *et al.* [16], focuses on best practices for individual hyperparameters rather than their interaction. Thus, driven by both the need for optimized RAG systems and the emergent challenge of evaluator unreliability, this research analyzes both the generation pipeline and its automated evaluation mechanism.

Roadmap

The subsequent sections of this report will elaborate on these investigations. Section 2 provides a literature review, establishing the academic context for LLMs, RAG, and evaluation methodologies. Section 3 details the datasets, the experimental setup for tuning the RAG pipeline, and the procedures for both automated and manual evaluation. Section 4 presents the core findings, analyzing the impact of hyperparameters on RAG accuracy (based on human judgments) and assessing the reliability of the LLM evaluator. Finally, Section 5 discusses the implications of these results, acknowledges the study's limitations, and suggests directions for future research.

2. Literature Review

From Statistical models to Transformers

The quest to enable machines to process and generate human language has deep roots in computational linguistics [1], [2]. Early approaches relied heavily on statistical language models, often based on Markov assumptions, predicting subsequent words based on preceding ones using probability [2], [17]. While foundational, these models treated language largely as symbol sequences, lacking deeper semantic understanding and struggling with long-range dependencies

[17]. The advent of using neural networks in the field of computational linguistics, particularly pioneered by Bengio *et al.*'s 2003 paper, marked a shift towards connectionist approaches [18]. As more powerful computing hardware became available and the field of neural networks evolved, recurrent neural networks and multi-layer perceptron models became prominent, improving predictive power [2], [19]. A significant leap occurred with Mikolov *et al.*'s work on word embeddings, which represented words as dense vectors in a continuous space where semantic similarity translated to proximity [19], [20]. This captured word relationships far more effectively than atomic representations and at lower computational cost than previous methods [20].

The introduction of the Transformer architecture by Vaswani *et al.* revolutionized the field [21]. Relying solely on self-attention mechanisms, Transformers could process entire sequences simultaneously, capturing complex dependencies between words regardless of their distance, proving more efficient and powerful than RNNs for many tasks [21]. Building on this, Devlin *et al.* introduced BERT (Bidirectional Encoder Representations from Transformers), achieving state-of-the-art results across numerous NLP benchmarks and solidifying the Transformer's dominance [22]. Consecutive research demonstrated that scaling model's like BERT yielded significant qualitative improvements ushering in the era of Large Language Models (LLMs) [2], [23].

The promise of LLMs and the need of Augmentation

Today, models like GPT-4, Google's Gemini, Claude, and Meta's LLaMa series represent the cutting edge, capable of sophisticated text generation, translation, and reasoning, arguably achieving the goal mentioned above, the goal laid out by none other than Alan Turing, the father

of theoretical computer science [1]. Many of the organizations behind the cutting-edge models are now discussing the possibilities and timelines of creation of Artificial General Intelligence (AGI) as research continues down this path [24], [2]. AGI refers to an artificial intelligence model which is expected to exceed human intelligence in every aspect [25]. However, these powerful models are not without limitations critical to practical deployment. Some of the key limitations of LLMs, as relevant to this study are:

1. Computational Cost & Knowledge Cutoff: Training state-of-the-art LLMs requires immense computational resources, leading to significant financial and environmental costs [26], [27], [28]. This cost also makes frequent retraining impractical, resulting in models having a fixed knowledge cutoff date, unable to access or reason about more recent information [26].
2. LLM Hallucinations: LLMs are prone to generating "hallucinations", which are confident yet factually incorrect or nonsensical statements [6], [29], [30]. The fluency of these outputs can make errors difficult to detect, posing risks if used in decision-making contexts [31].
3. Knowledge Recall: LLMs may struggle to reliably recall or utilize less common ("long-tail") information present in their vast training data [4], [5].

Retrieval-Augmented Generation (RAG) as a Solution

Retrieval-Augmented Generation (RAG) has emerged as a prominent technique to address these LLM limitations [8], [26]. A typical RAG pipeline involves [9], [8]:

- Retrieval: Given a user query, relevant information is retrieved from an external knowledge source.

- Augmentation: The retrieved information is processed and synthesized, often alongside the original query, to form a rich prompt for the LLM.
- Generation: The LLM uses the augmented prompt to generate a final response, grounding its output in the retrieved external knowledge.

By incorporating external, often curated and up-to-date information, RAG aims to reduce hallucinations, overcome knowledge cutoffs, and improve factual accuracy [6], [11]. It allows LLMs to leverage domain-specific or proprietary data securely without fine-tuning the base model. Key hyperparameters governing the retrieval process include chunk size (the size of text segments in the knowledge base) and top_k (the number of retrieved segments provided to the LLM). The choice of these parameters significantly impacts the context provided to the LLM, influencing the relevance and completeness of the final answer. While studies like Wang *et al.* [16] offer valuable best practices for individual RAG components, research systematically exploring the combined effect of chunk size and top_k, particularly for specific model families like LLaMa, remains less common.

Evaluating RAG Systems and the Challenge of Automated Assessment

Evaluating the performance of RAG systems typically involves using question-answering datasets or other NLP benchmarks [32]. Output quality can be assessed using various methods:

- Lexical Overlap Metrics: ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [33] and BLEU (Bilingual Evaluation Understudy) [34] measure n-gram overlap between generated and reference texts.
- Semantic Similarity Metrics: BertScore [35] uses contextual embeddings to compare semantic similarity.

- LLM-based Evaluation: Using another powerful LLM (e.g., GPT-4) to judge the quality, coherence, or factual accuracy of the generated output [10], [11].

This study initially employed the LLM-based evaluation approach, aiming for scalability. However, the documented challenges of LLM "hallucinations" [29] extend to their use as evaluators. Ensuring the reliability and consistency of LLM-based judgments is an active area of research [15]. Concerns exist about whether LLMs can robustly follow evaluation criteria and whether their assessments align well with human judgment, especially given their own inherent biases and potential inaccuracies [15]. Existing literature highlights the need for caution and often recommends human validation [15], yet systematic comparisons between LLM evaluators and human judgments under varying RAG configurations are still needed.

This Study's Contribution

This research addresses two key areas identified in the literature:

1. It provides an empirical analysis of the interplay between chunk size and top_k hyperparameters on the performance of a LLaMa 8B-based RAG pipeline, offering insights beyond individual parameter tuning.
2. It directly confronts the challenge of automated evaluation by comparing an LLM evaluator's reliability against human judgment across diverse RAG configurations, quantifying the alignment and potential discrepancies.

By investigating these aspects using a Wikipedia-based QA dataset predating the LLM era [13] (mitigating potential data contamination risks) and employing rigorous manual evaluation alongside automated methods, this study aims to contribute practical guidelines for

configuring RAG systems and shed light on the current limitations and potential of using LLMs for automated evaluation tasks.

3. Dataset

The dataset utilized for this project originates from a publicly available Wikipedia question-answer dataset released under the CC BY-SA 3.0 license. The data was collected between 2008 and 2010 by staff and students at Carnegie Mellon University and the University of Pittsburgh. A subset of this data exists on Hugging Face, titled rag-mini-wikipedia, which includes both the question-answer component and the raw text corpus from Wikipedia. This dataset is well-suited for this project as it provides a reliable and structured corpus for building the knowledge base, along with an associated set of question-answer pairs for testing and evaluation.

Attributes of the dataset

The dataset has two components:

1. The question – answer dataset:

Question	**Answer**
When did Lincoln begin his political career?	1832
What did The Legal Tender Act of 1862 establish?	the United States Note, the first paper currency in United States history

2. The raw text corpus:

Passage
Faraday later used the principle to construct the electric dynamo, the ancestor of modern power generators.
In 1839 he completed a series of experiments aimed at investigating the fundamental nature of electricity. Faraday used "static", batteries, and "animal electricity" to produce the phenomena of electrostatic attraction, electrolysis, magnetism, etc. He concluded that, contrary to scientific opinion of the time, the divisions between the various "kinds" of electricity were illusory. Faraday instead proposed that only a single "electricity" exists, and the changing values of quantity and intensity (voltage and charge) would produce different groups of phenomena.

The metadata of the dataset is like so:

Component	**Details**
Question	918 rows
Answer	918 rows
Passage	3,200 rows

For the construction of the knowledge base, the 3,200 rows of text will be concatenated into one large text. This large text will then be broken into smaller chunks according to the chunk size parameter in the RAG pipeline.

Assumption about parameters

A core assumption is the accuracy of the question-answer pairings provided within the dataset. It is further assumed that the necessary information to answer each question is contained within the provided raw text corpus.

4. Methodology

This study employs a multi-stage methodology designed to systematically evaluate the impact of hyperparameters (chunk_size, top_k) on a Retrieval-Augmented Generation (RAG) pipeline's performance and critically assess the reliability of an automated Large Language Model (LLM) based evaluator. The initial approach focused on automated execution and evaluation, but preliminary findings necessitated the incorporation of a manual evaluation phase to establish a reliable benchmark. This section details the methods used in each stage.

Automated RAG Pipeline and Initial Evaluation

The primary goal of this stage was to generate output from the RAG pipeline across a predefined grid of hyperparameter configurations and obtain an initial performance assessment using an LLM-based evaluator. An automated RAG pipeline coupled with an LLM evaluator was initially chosen for its scalability and efficiency. This approach allowed for the systematic testing of 40 different hyperparameter combinations (chunk_size * top_k) across a substantial dataset (918 questions) within a manageable timeframe (60 GPU hours), which would be infeasible to evaluate manually in its entirety. Standard open-source libraries like Hugging Face Transformers and Haystack provide robust frameworks for building and executing such pipelines.

The process was implemented using a Python script leveraging the Haystack AI framework and Hugging Face's Transformers library, detailed in the accompanying code notebook. The foundation of the process was the rag-datasets/rag-mini-wikipedia dataset from Hugging Face, which supplied 918 question-answer pairs for testing and 3,200 text passages for the knowledge base. Preparation of the knowledge base began with preprocessing the text passages using basic cleaning steps via the preprocess_text function.

These cleaned passages were then concatenated into a single corpus. Subsequently, this corpus was segmented into overlapping chunks according to the chunk_size hyperparameter (testing values of 50, 100, 150, and 200 words) with a fixed overlap of 20 words, using the split_content_into_chunks function. For each chunk_size, an InMemoryDocumentStore was prepared, with caching employed (document_stores_cache) to optimize repeated runs. These document chunks were then embedded using SentenceTransformersDocumentEmbedder with the sentence-transformers/all-MiniLM-L6-v2 model (384 dimensions) and stored in their corresponding document store; embedding was performed in batches on the GPU for efficiency.

Within the RAG pipeline itself, incoming questions were first embedded using SentenceTransformersTextEmbedder with the same all-MiniLM-L6-v2 model used for documents. An InMemoryEmbeddingRetriever then identified the top_k most relevant document chunks (testing values from 1 to 10) from the appropriate document store based on cosine similarity between the question embedding and document embeddings. For the generation step, a meta-llama/Llama-3.1-8B model served as the generator, loaded with 8-bit quantization to manage memory constraints. The ModelHandler class managed model loading and text generation via the Transformers pipeline, utilizing a specific prompt template that provided the

retrieved context and the original question to guide the LLM towards a concise answer (setting max_tokens to 10 and temperature to 0 for deterministic output).

Concurrently, an automated evaluator model was employed. Although initially planned to be a Llama 3.2 1B model, this was upgraded to another instance of meta-llama/Llama-3.1-8B (also 8-bit quantized) because the smaller model struggled to follow instructions reliably. This evaluator received a prompt asking for a simple "Yes" or "No" judgment on the semantic equivalence between the RAG-generated answer and the ground truth answer, considering the original question (using max_tokens=3 and temperature=0).

The core of the automated phase involved an experiment loop where the script iterated through all 40 combinations of chunk_size and top_k, processing all 918 questions for each unique configuration. For every experiment run, the output was saved to a CSV file containing the question, the ground truth answer, the RAG pipeline's generated answer, and the raw output from the evaluator LLM.

This automated approach was chosen because it allowed for systematic exploration of the hyperparameter space more efficiently than manual methods, leveraging state-of-the-art open-source models and frameworks. However, it necessitated significant computational resources (covered by Kaggle GPU access) and its results were inherently dependent on the reliability of the LLM evaluator, which was initially an unknown factor. Furthermore, this pipelined approach carried a risk of cascading inaccuracies if any single component performed poorly. Alternative strategies were considered but deemed less suitable.

Full manual evaluation of all 36,720 generated answers (40 experiments * 918 questions) was rejected due to the prohibitive time investment. Using simpler lexical overlap metrics like

BLEU or ROUGE was considered faster but potentially inadequate for capturing the target criterion of semantic equivalence, as these metrics can penalize correct but differently phrased answers or reward incorrect answers that merely reuse context words. Finally, employing proprietary LLM APIs like GPT-4 for evaluation was avoided as it would introduce external dependencies, potential costs, less control, and confound the analysis by using a potentially larger and different model than the LLaMa-based RAG system under investigation.

Assessment of Evaluator's Reliability and Manual Evaluation

Early in the experimentation process, it was found that the LLM evaluator model was incorrectly classifying the evaluation via spot checks. However, due to the time and resource constraints limiting the study, the decision was made to continue the experiments.

Question	**Generated Answer**	**Actual Answer**	**LLM Evaluation**	**Human Evaluation**
Who got Seward elected to the senate?	Weed ultimately got Seward elected to the senate.	Weed.	No	Yes
These projections, called papillae, have what?	These projections, called papillae, have what	a rich blood supply	a rich	No
Where was Grover	Grover Cleveland was	In the White House	No	Yes

Cleveland married?	married in the White House.			

Table 2: Examples from manual human evaluation. These examples illustrate the unreliability of the LLM Evaluator. The LLM Evaluation is based on the question "Is the semantic meaning of the generated the same as the truth?". Whereas the human evaluation is based on the question "Does the generated answer semantically match the ground truth, considering the original question?"

The types of erroneous evaluations ranged from incorrect evaluations to formatting errors, to repeating portions of the generated_answer or the ground_truth. 30 fixed random samples were taken from the 40 experiments to create a new document of 1200 evaluations. In an initial analysis of the sample evaluations, 6 out of the 1200 evaluations were incorrectly formatted but followed the instructions of replying in either "Yes" or "No", while 3 of the evaluations were random words.

The 1200 evaluations were then manually evaluated by the author where, discarding the LLM evaluator's evaluations, each RAG generated answer was compared to the ground truth to determine the accuracy of the RAG pipeline's answer.

It was found that across the 1200 samples, the LLM evaluator had an accuracy of 73.4%. This metric demonstrated the unreliability of the automated LLM evaluator for the task. While it was a setback, this finding also provided an opportunity to expand the scope of the initial study to include analysis of the reliability of LLM evaluators. The 1200 manual evaluations thus provided a dataset for:

- Reliably assess the impact of chunk_size and top_k on RAG performance.

- Quantitatively measuring the automated evaluator's reliability.

The 30 fixed random samples were generated using Python's NumPy library's sample function with a random_state of 42. 30 samples were chosen to provide a reasonable balance between obtaining a robust representation of the experiment and the time it would take to manually evaluate the 1200 samples. The manual evaluation was done by asking the question "Does the generated answer semantically match the ground truth, considering the original question?". "yes" was labeled if the human evaluator determined it to be true, and "no" otherwise. Minor differences in phrasing and punctuation were ignored. Table 2 illustrates some examples of the erroneous classification that were done by the LLM evaluator.

Data and Statistical Analysis

The analysis phase aimed to address the two primary research questions using the dataset derived from the 1200 manually evaluated samples, which included the human labels, the automated evaluator's raw output, and the corresponding hyperparameter settings (chunk_size, top_k). All analyses were conducted using Python with the pandas, numpy, statsmodels, scikit-learn, matplotlib, and seaborn libraries.

First, descriptive statistics and visualizations were employed to understand initial patterns. For each of the 40 experimental conditions (chunk_size x top_k), the mean accuracy and standard error were calculated based on the human evaluations (human_binary). Similarly, the mean agreement rate between the human and the (cleaned) automated evaluator (eval_accuracy) was calculated per condition. Visualizations, including heatmaps (plots 1.3, 2.4) and line plots (plot 1.4), were generated to illustrate the relationship between the hyperparameters and both human-rated accuracy and evaluator agreement. A confusion matrix

(plot 2.5) was generated to detail the types of agreement and disagreement between the human rater and the automated evaluator. Overall reliability metrics, including percentage agreement and Cohen's Kappa, were calculated for the automated evaluator based on the 1200 samples.

Second, inferential statistical modeling was used to rigorously test the hypotheses related to the research questions. To assess the effect of chunk_size and top_k on the semantic accuracy of the RAG pipeline, a logistic regression model was fitted using the statsmodels library. The dependent variable was the binary human rating (human_binary: 1=correct, 0=incorrect). The independent variables were chunk_size (treated as a categorical variable with 50 as the baseline), top_k (treated as a continuous variable), and their interaction term (C(chunk_size):top_k). This method was chosen for its suitability in modeling binary outcomes and its ability to test the significance and direction of the main effects of each hyperparameter and their potential interaction, controlling for the other variables.

To investigate whether the experimental conditions influenced the reliability of the automated LLM evaluator, a second logistic regression model was fitted. The dependent variable was eval_accuracy (1=evaluator agreed with human, 0=disagreed). The independent variables were identical to the first model: chunk_size (categorical, 50 baseline), top_k (continuous), and their interaction term. This analysis directly tested if changes in chunk size or the number of retrieved documents systematically affected the likelihood of the evaluator providing a judgment consistent with the human standard.

For both models, statistical significance was assessed using a conventional alpha level of 0.05. Model summaries, including coefficients (log-odds), standard errors, z-scores, p-values, and confidence intervals, were used for interpretation.

5. Results

Prior to analyzing the main research questions, interaction models including chunk_size, top_k, and their interaction term were compared against main effect models (including only chunk_size and top_k) for both Research Question 1 (RQ1: Human Accuracy) and Research Question 2 (RQ2: Evaluator Agreement). Log-likelihood ratio tests revealed no statistically significant difference between the interaction and main effect models for either RQ1 ($p = 0.896$) or RQ2 ($p = 0.433$). Therefore, the more parsimonious main effect models were selected for subsequent analysis and interpretation.

RAG Pipeline Performance (Human Evaluation)

The analysis of the 1200 manually evaluated samples revealed that the semantic accuracy of the Llama 3.1 8B-based RAG pipeline varied considerably depending on the chunk_size and top_k configuration. Overall human-rated accuracy ranged across the 40 conditions. The optimal performance, based on the highest mean accuracy in the evaluated samples, was observed for the configuration with chunk_size = 100 and top_k = 1, achieving 70.0% accuracy.

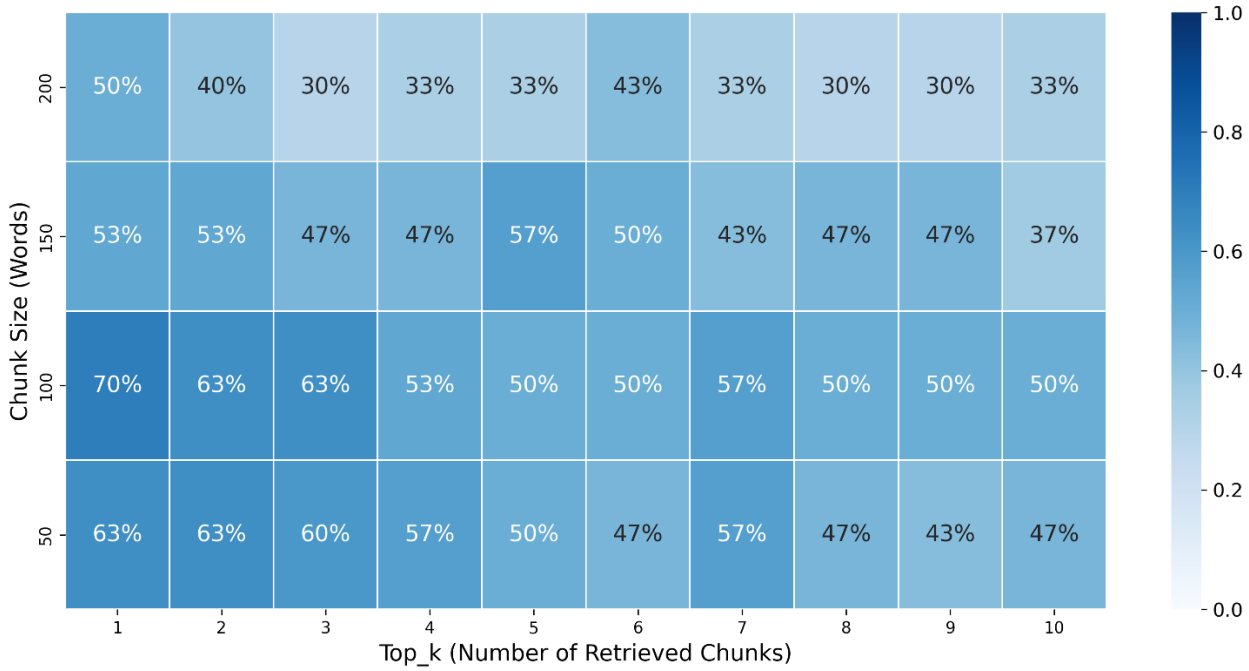

Figure 2: Heatmap of Human-Rated RAG Accuracy (%) vs. Chunk Size and Top_k.

The relationship between hyperparameters and human-rated accuracy is visualized in Figure 2 and Figure 3. The heatmap (Figure 2) provides an overview, indicating generally higher accuracy at smaller top_k values, particularly for chunk sizes 50 and 100. The line plot (Figure 3) clearly illustrates a general trend: for all chunk sizes, accuracy tended to decrease as top_k increased beyond the initial values. The performance for chunk sizes 50 and 100 peaked at top_k=1, while chunk sizes 150 and 200 exhibited lower overall accuracy and a less distinct peak.

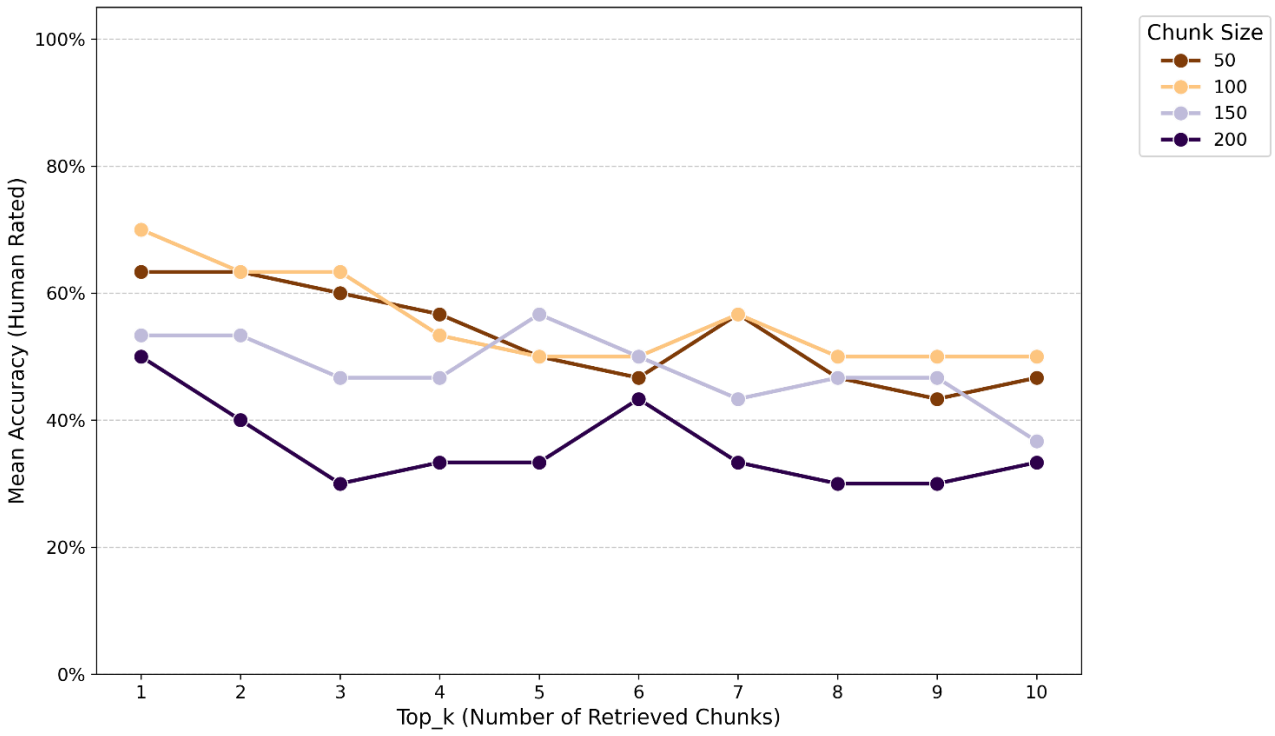

Figure 3: Human-Rated RAG Accuracy vs. Top_k for different Chunk Sizes.

Formal testing via logistic regression confirmed these observations (Table 3). The overall model was statistically significant (LLR $p < 0.001$), indicating that the hyperparameters collectively influence performance. The model results are summarized below:

- Chunk Size: Compared to the baseline chunk size of 50, using chunk_size = 200 resulted in a statistically significant decrease in the log-odds of achieving a correct answer (coef = -0.7304, $p < 0.001$). The log-odds for chunk_size = 100 (coef = 0.0950, $p = 0.564$) and chunk_size = 150 (coef = -0.2157, $p = 0.189$) were not significantly different from the baseline (chunk_size = 50). This suggests that significantly larger chunk sizes (200 words) negatively impact the RAG pipeline's ability to generate accurate answers in this setup.

- Top_k: The parameter had a statistically significant negative effect on the log-odds of a correct answer (coef = -0.0696, p = 0.001). This indicates that retrieving more documents (increasing top_k) significantly reduced the likelihood of generating a semantically correct answer, controlling for chunk size.

Predictor	**Coefficient**	**Standard Error (SE)**	**z-value**	**p-value**	**[0.025, 0.975] Confidence Interval**
Intercept	0.5179	0.163	3.183	0.001	0.199 to 0.837
C(chunk_size_cat)[T.100]	0.0950	0.165	0.577	0.564	-0.228 to 0.418
C(chunk_size_cat)[T.150]	-0.2157	0.164	-1.312	0.189	-0.538 to 0.106
C(chunk_size_cat)[T.200]	-0.7304	0.168	-4.349	0.000	-1.060 to -0.401
top_k	-0.0696	0.021	-3.390	0.001	-0.110 to -0.029

Table 3: Logistic Regression Results for Human Evaluation Accuracy. This table presents the logistic regression model assessing how the hyperparameters (chunk_size and top_k) affect the probability that the human evaluator agrees with the generated answer. The dependent variable (human_binary) indicates whether the generated answer was deemed correct (1 = Yes, 0 = No). Coefficients, standard errors, z-values, and significance levels are reported for each predictor.

In summary, for the first research question, the results based on the main effects model indicate that using a larger chunk size (specifically 200) and retrieving more documents

(increasing top_k) were significantly associated with lower human-rated accuracy for this RAG setup. Optimal performance tended towards smaller chunk sizes and retrieving fewer documents.

Automated Evaluator Reliability

The assessment of the LLM evaluator focused on its agreement with the human evaluator across 1200 samples. The overall agreement rate was found to be 73.4%. Cohen's Kappa, which corrects for chance agreement, was calculated as $\kappa = 0.47$, suggesting moderate agreement between the automated evaluator and the human standard.

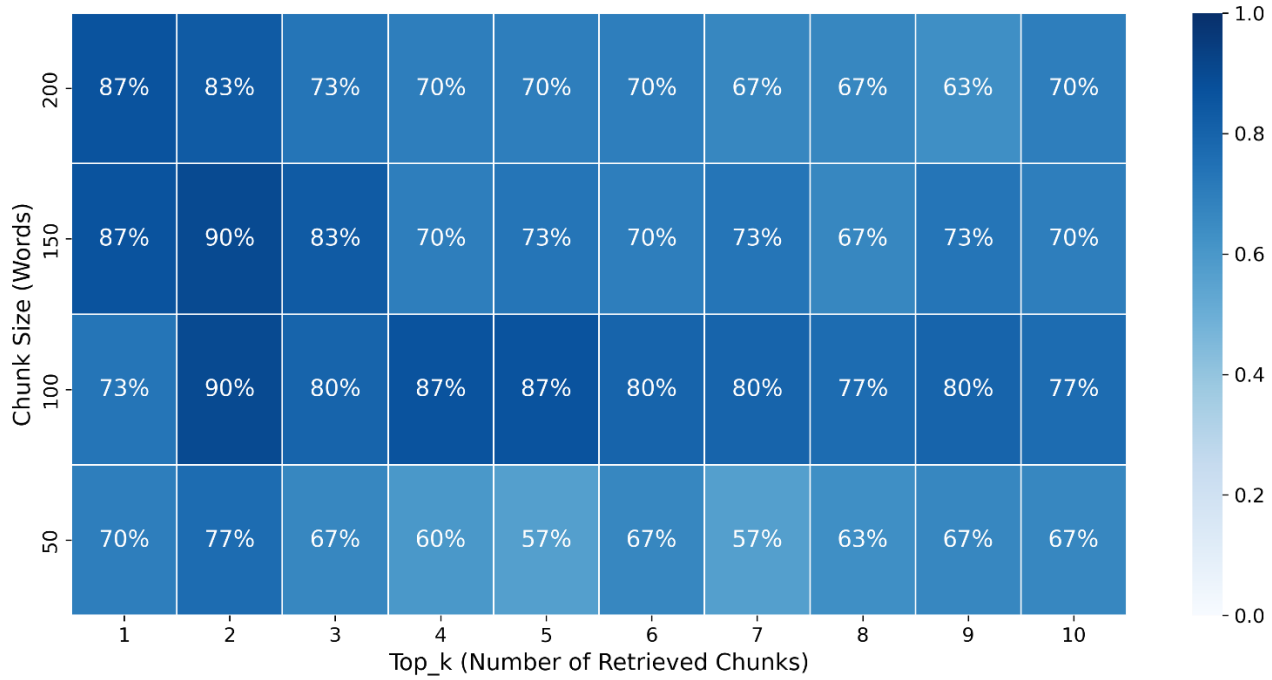

Figure 4: Heatmap of Agreement Rate (%) (Evaluator vs Human) vs. Chunk Size and Top_k.

The variation in agreement across different experimental conditions is shown in Figure 4. Visual inspection suggests higher agreement rates for intermediate chunk sizes (100, 150). The nature of disagreements is detailed in the confusion matrix presented in Figure 5. Out of the 1200 samples, there were 254 False Positives (evaluator incorrectly labelled 'Yes') and 65 False

Negatives (evaluator incorrectly labelled 'No'), indicating specific biases or failure modes in the evaluator's judgments.

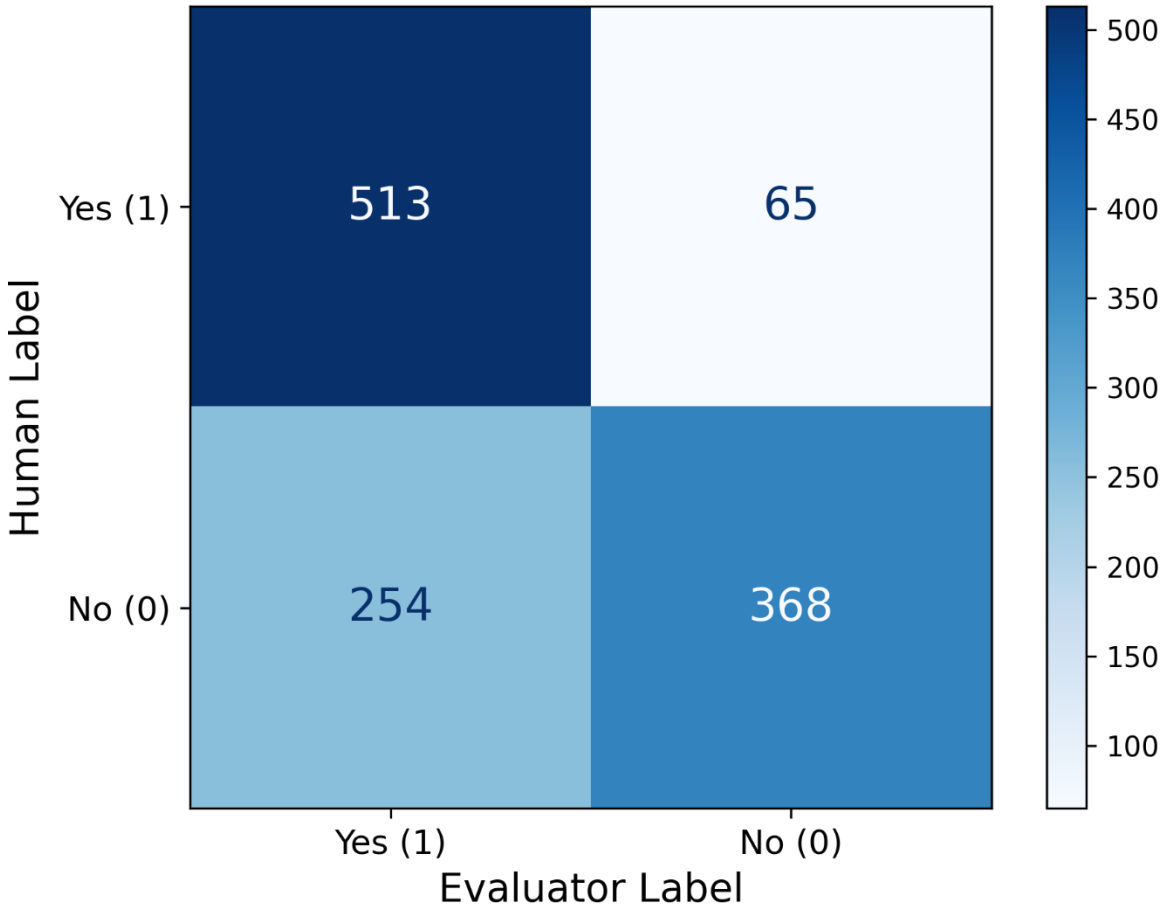

Figure 5: Confusion Matrix: Human Label vs. Evaluator Label.

The logistic regression analysis investigating factors influencing evaluator agreement (Table 4) revealed that chunk_size significantly impacted reliability, while top_k did not. The overall model was statistically significant (LLR $p < 0.001$), suggesting these hyperparameters influence evaluator reliability. The model results are summarized below:

- Chunk_size: Compared to the baseline chunk size of 50, the odds of the evaluator agreeing with the human were significantly higher for chunk_size = 100 (coef =

0.8374, $p < 0.001$) and chunk_size = 150 (coef = 0.5197, $p = 0.004$). The difference for chunk_size = 200 (coef = 0.3282, $p = 0.064$) was not statistically significant at the alpha=0.05 level compared to the baseline chunk size of 50. This suggests the evaluator's reliability was lowest at the smallest chunk size (50) and improved significantly at intermediate sizes (100, 150).

- Top_k: The top_k parameter had a statistically significant negative effect on the log-odds of evaluator agreement (coef = -0.0697, $p = 0.003$). This implies that as more documents were retrieved and presumably included in the context for the generation (and potentially influencing evaluation context indirectly, though the evaluator primarily saw generated vs. ground truth), the likelihood of the automated evaluator agreeing with the human decreased significantly.

Predictor	**Coefficient**	**Standard Error (SE)**	**z-value**	**p-value**	**[0.025, 0.975] Confidence Interval**
Intercept	1.0084	0.179	5.632	0.000	0.657 to 1.359
C(chunk_size_cat)[T.100]	0.8374	0.191	4.377	0.000	0.462 to 1.212
C(chunk_size_cat)[T.150]	0.5197	0.182	2.860	0.004	0.164 to 0.876
C(chunk_size_cat)[T.200]	0.3282	0.177	1.851	0.064	-0.019 to 0.676

top_k	-0.0697	0.023	-3.008	0.003	-0.115 to -0.024

Table 4: Logistic Regression for Evaluator Correctness (Agreement). This table details the logistic regression analysis that examines how hyperparameters (chunk_size and top_k) influence the likelihood that the LLM evaluator agrees with the human evaluation (eval_accuracy). Coefficients, their standard errors, z-values, p-values, and 95% confidence intervals are reported.

Therefore, for the second research question, the findings indicate that the reliability of the Llama 3.1 8B evaluator was influenced by both chunk_size and top_k. Agreement was lowest at the smallest chunk size (50) and decreased as more documents were retrieved (top_k increased).

6. Discussion and Conclusion

This study investigated the dual challenge of optimizing Retrieval-Augmented Generation (RAG) hyperparameters and evaluating the reliability of automated LLM-based assessment. The results provide practical insights into configuring RAG systems using models like Llama 3.1 8B and highlight crucial considerations for using LLMs as evaluators.

Impact of Hyperparameters on RAG Accuracy

The findings regarding research question 1 strongly suggest that, for the specific RAG pipeline and Llama 3.1 8B model (8-bit quantized) used in this study, "less is more" in terms of retrieved context. Both increasing the number of retrieved chunks (top_k) and using a significantly larger chunk size (200 words) were detrimental to generating semantically accurate answers compared to the ground truth. The optimal performance trended towards smaller chunk sizes (like 50 or 100 words, which were not significantly different from each other) combined with retrieving only the single most relevant chunk (top_k=1).

This aligns with the intuition that providing excessive or less relevant context can confuse the LLM or dilute the impact of the most pertinent information. While RAG aims to provide necessary external knowledge, bombarding the LLM with too many chunks (higher top_k) or overly long chunks that might contain mixed relevance (potentially larger chunk_size) can hinder its ability to synthesize a precise answer. This contrasts somewhat with approaches that might assume more context is always better and underscores the need for empirical tuning rather than relying solely on intuition. The lack of significant interaction effects suggests these detrimental effects acted somewhat independently within the tested ranges.

Reliability of Automated LLM Evaluation

The second major finding concerns the reliability of using an LLM (in this case, the same Llama 3.1 8B model) as an automated evaluator for semantic equivalence. With an overall agreement of 73.4% and a moderate Kappa score (0.47), the evaluator demonstrated a level of utility but fell significantly short of perfect alignment with human judgment. This empirically validates concerns raised in the literature about the trustworthiness of LLM-based evaluation [15]. The evaluator was particularly prone to false positives (incorrectly agreeing with the generated answer), which could be problematic in scenarios requiring high precision.

Interestingly, the evaluator's reliability was itself influenced by the RAG hyperparameters. Agreement was lowest when the RAG pipeline used the smallest chunk size (50) and decreased as top_k increased. The lower agreement at chunk_size=50 might indicate the evaluator struggled more when the generated answers were based on very short context pieces. The negative impact of top_k on agreement is harder to interpret directly, as the evaluator primarily saw the question, generated answer, and ground truth, but perhaps the nature of

answers generated with higher top_k values (which were also less accurate according to the human) presented more challenging or ambiguous cases for the automated evaluator. This finding implies that the reliability of an LLM evaluator might not be constant but can vary depending on the characteristics of the system generating the output being evaluated.

Implications and Limitations

The practical implication for RAG system builders using similar setups is to carefully tune chunk_size and top_k, likely favoring smaller values, and to rigorously validate performance against human judgment rather than solely relying on automated metrics or LLM evaluators. The moderate reliability of the LLM evaluator suggests it might be useful for preliminary checks or trend analysis but cannot fully replace human oversight for critical applications.

This study has several limitations. First, it used a specific LLM (Llama 3.1 8B, 8-bit quantized) for both generation and evaluation; results might differ with other models, architectures, or quantization levels. Second, the findings are based on a single dataset (rag-mini-wikipedia) [13] and evaluation task (semantic equivalence); generalizability to other domains or tasks requires further investigation. Third, the manual evaluation was performed by a single human researcher, introducing potential subjectivity, although efforts were made to apply the criteria consistently. Finally, other hyperparameters of the RAG system (e.g., embedding model, prompt templates, generator settings like temperature) were held constant and could also interact with chunk_size and top_k.

Future work

This study contributes empirical evidence on optimizing key RAG hyperparameters and quantifies the reliability challenge inherent in using current LLMs for automated evaluation. It underscores the continued need for human oversight in deploying and assessing AI systems, particularly in applications demanding high accuracy and trustworthiness.

Future research could expand upon these findings in several directions. Investigating these hyperparameter effects across different LLM architectures (e.g., GPT, Gemini, Claude) and sizes, as well as different embedding models, would enhance generalizability. Exploring more sophisticated retrieval strategies beyond simple top_k selection, potentially involving re-ranking or context distillation, could yield performance improvements. Replicating the evaluation reliability analysis with multiple human evaluators and potentially more advanced LLM evaluators (like GPT-4 or specialized evaluation models) would provide a more robust assessment. Finally, extending the analysis to different datasets, domains (e.g., enterprise knowledge bases), and downstream tasks beyond question-answering would broaden the applicability of the findings.

References

[1] A. Turing, “Computing machinery and intelligence,” *Mind*, vol. 59, no. 236, pp. 433–60, 1950.

[2] W. X. Zhao *et al.*, “A Survey of Large Language Models.” 2025. [Online]. Available: https://arxiv.org/abs/2303.18223

[3] J. Sommer, “Is artificial intelligence really worth the hype?,” Feb. 2025, [Online]. Available: https://www.nytimes.com/2025/02/07/business/ai-deepseek-nvidia-tesla.html

[4] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, “Large Language Models Struggle to Learn Long-Tail Knowledge.” 2023. [Online]. Available: https://arxiv.org/abs/2211.08411

[5] Y. Gao *et al.*, “Retrieval-Augmented Generation for Large Language Models: A Survey.” 2024. [Online]. Available: https://arxiv.org/abs/2312.10997

[6] Y. Zhang *et al.*, “Siren’s Song in the AI Ocean: A Survey on Hallucination in Large Language Models.” 2023. [Online]. Available: https://arxiv.org/abs/2309.01219

[7] K. Buchholz, “The Extreme Cost Of Training AI Models,” *Forbes*, Aug. 2024, [Online]. Available: https://www.forbes.com/sites/katharinabuchholz/2024/08/23/the-extreme-cost-of-training-ai-models/

[8] P. Lewis *et al.*, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.” 2021. [Online]. Available: https://arxiv.org/abs/2005.11401

[9] A. Singh, A. Ehtesham, S. Kumar, and T. T. Khoei, “Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG.” 2025. [Online]. Available: https://arxiv.org/abs/2501.09136

[10] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, “G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment.” 2023. [Online]. Available: https://arxiv.org/abs/2303.16634

[11] P. Finardi *et al.*, “The Chronicles of RAG: The Retriever, the Chunk and the Generator.” 2024. [Online]. Available: https://arxiv.org/abs/2401.07883

[12] H. Touvron *et al.*, “LLaMA: Open and Efficient Foundation Language Models.” 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[13] N. Smith, M. Heilman, and R. Hwa, “Question Generation as a Competitive Undergraduate Course Project,” Jan. 2008.

[14] K. S. Adebayo, “Is Agentic AI ready to handle the way we do business?” Mar. 2025. [Online]. Available: https://www.forbes.com/sites/kolawolesamueladebayo/2025/03/17/is-agentic-ai-ready-to-handle-the-way-we-do-business/

[15] S. Brodsky, “Who watches the AI watchers? The challenge of self-evaluating AI.” Nov. 2024. [Online]. Available: https://www.ibm.com/think/news/ai-testing-advances

[16] X. Wang *et al.*, “Searching for Best Practices in Retrieval-Augmented Generation.” 2024. [Online]. Available: https://arxiv.org/abs/2407.01219

[17] R. Rosenfeld, “Two decades of statistical language modeling: where do we go from here?,” *Proc. IEEE*, vol. 88, no. 8, pp. 1270–1278, 2000.

[18] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A Neural Probabilistic Language Model,” *J. Mach. Learn. Res.*, vol. 3, no. 6, pp. 1137–1155, 2003.

[19] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, “Recurrent neural network based language model.,” in *Interspeech*, Makuhari, 2010, pp. 1045–1048.

[20] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 142–150.
[21] A. Vaswani *et al.*, “Attention Is All You Need.” 2023. [Online]. Available: https://arxiv.org/abs/1706.03762
[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” 2019. [Online]. Available: https://arxiv.org/abs/1810.04805
[23] M. Shanahan, “Talking About Large Language Models.” 2023. [Online]. Available: https://arxiv.org/abs/2212.03551
[24] R. Shah *et al.*, “An Approach to Technical AGI Safety and Security,” *ArXiv Prepr. ArXiv250401849*, 2025.
[25] S. McLean, G. J. M. Read, J. Thompson, C. Baber, N. A. Stanton, and P. M. S. and, “The risks associated with Artificial General Intelligence: A systematic review,” *J. Exp. Theor. Artif. Intell.*, vol. 35, no. 5, pp. 649–663, 2023, doi: 10.1080/0952813X.2021.1964003.
[26] H. Naveed *et al.*, “A comprehensive overview of large language models,” *ArXiv Prepr. ArXiv230706435*, 2023.
[27] E. Strubell, A. Ganesh, and A. McCallum, “Energy and Policy Considerations for Deep Learning in NLP.” 2019. [Online]. Available: https://arxiv.org/abs/1906.02243
[28] J. Stojkovic, E. Choukse, C. Zhang, I. Goiri, and J. Torrellas, “Towards Greener LLMs: Bringing Energy-Efficiency to the Forefront of LLM Inference.” 2024. [Online]. Available: https://arxiv.org/abs/2403.20306
[29] N. M. Guerreiro *et al.*, “Hallucinations in Large Multilingual Translation Models.” 2023. [Online]. Available: https://arxiv.org/abs/2303.16104
[30] Y. Bang *et al.*, “A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity.” 2023. [Online]. Available: https://arxiv.org/abs/2302.04023
[31] L. Huang *et al.*, “A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions,” *ACM Trans. Inf. Syst.*, vol. 43, no. 2, pp. 1–55, Jan. 2025, doi: 10.1145/3703155.
[32] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu, and Z. Liu, “Evaluation of Retrieval-Augmented Generation: A Survey,” in *Big Data*, Springer Nature Singapore, 2025, pp. 102–120. doi: 10.1007/978-981-96-1024-2_8.
[33] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” in *Text Summarization Branches Out*, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013/
[34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a Method for Automatic Evaluation of Machine Translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, P. Isabelle, E. Charniak, and D. Lin, Eds., Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. doi: 10.3115/1073083.1073135.
[35] T. Zhang*, V. Kishore*, F. Wu*, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating Text Generation with BERT,” in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr